UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Modelado indirecto de usuarios de un bot conversacional

Autor: Guillermo Torres Zamora
Tutor: Álvaro Ortigosa

febrero 2021

Guillermo Torres Zamora
Modelado indirecto de usuarios de un bot conversacional

Guillermo Torres Zamora
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

Resumen

Conocer la personalidad de un usuario puede resultar de mucha utilidad en diferentes contextos de sistemas adaptativos y de recomendación. Sin embargo, averiguar la personalidad de un usuario mediante cuestionarios puede resultar muy intrusivo y propenso a errores.

Una alternativa es realizar un perfilado indirecto de personalidad, observando el comportamiento de los usuarios en diferentes contextos como, por ejemplo, en una conversación.

Por ello, en este trabajo se ha desarrollado la infraestura software necesaria para recabar los datos y obtener la personalidad del usuario de un bot conversacional a partir de como interactua con dicho bot.

Primero se ha diseñado un bot conversacional para Telegram que realiza un test de personalidad y una serie de preguntas respecto a los gustos de series de televisión.

Después se han recolectado datos mediante la promoción del bot conversacional a través de contactos conocidos. Y se ha diseñado e implementado una herramienta para analizar mediante aprendizaje automático, los datos recabados de las conversaciones de los usuarios con el bot.

Para satisfacer al usuario, se le da el resultado del test psicólogico y se le recomienda una serie de televisión dentro de las distintas plataformas de streaming. Esta característica, aún siendo secundaria, aumenta signficativamente la satisfación del usuario y la probabilidad de que lo recomiende a otros. Gracias a estas características hemos conseguido recabar datos de 433 personas.

En cuanto a los resultados del trabajo, se ha conseguido implementar la infraestructura con éxito y el bot ha conseguido funcionar sin problemas. El programa de análisis de datos ha logrado funcionar correctamente de manera automática, sin que se haya observado ningún fallo.

Aunque la infraestructura funciona correctamente no se ha podido obtener relación entre el uso del bot por parte del usuario y su personalidad. Hacen falta más trabajos para identificar la causa del problema y ver si este modelado es realmente posible.

Palabras clave

bot conversacional, personalidad, modelado indirecto

Abstract

Know the personality of a user could be very helpful in different contexts in adaptive systems and recommendation systems. However, find the personality of a user using surveys could be intrusive and error-prone.

An alternative is to make and indirect user modeling, observing the behavior of the user in different contexts. For example in a conversation.

Therefore, in this bachelor thesis, we have developed the software infrastructure needed to recolect the data and obtain the personality of the chatbot user using how he/she interacts with the chatbot.

Firstly, we have designed a Telegram chatbot that performs a personality test and asks the user a set of questions about his TV series tastes.

Secondly, we have collected the data through the promotion of the chatbot among our contact network. Also, we have developed and implemented a tool for analysing the data collected through the user's interactions with the chatbot.

For a better user satisfaction, the bot gives him/her the results of the psychological test and the recommendation of a TV series among those available on an array of streaming platforms. This secondary feature increases significantly the user satisfaction and increases the probability of him/her recommending the chatbot to other persons. Thanks to this feature we have collected data from 433 users.

Regarding the results of this work, we were able to successfully implement the infrastructure and the chatbot worked without any glitches. Also, the data analysis program was able to work automatically without any problem nor have we observed any malfunction.

Although the infrastructure works correctly, the analysis program has not been able to obtain any connection between the user's interaction with the chatbot and his/her personality. More research is needed to identify the root cause of this problem and to understand if this type of user modeling is in fact possible.

Keywords

chatbot, personality, indirect modeling

Índice

Listas

Lista de figuras

Lista de tablas

1

Introducción

En este trabajo se ha desarrollado un bot conversacional en la plataforma de mensajería Telegram con el objetivo de encontrar un modelo entre las interacciones de un usuario con un bot conversacional y su personalidad, para así poder en trabajos posteriores diseñar sistemas que se adapten al usuario y a sus necesidades.

Desde la antigüedad se ha intentado definir el término personalidad y nos encontramos con multitud de definiciones. La mayoría de las definiciones se basan en dos enfoques: en tipos de personalidad o en rasgos de personalidad. Por tipos, la persona pertenece a una categoría de personalidad concreta y por rasgos, cada individuo expresa una serie de rasgos en un determinado porcentaje. [1]

Nosotros hemos optado por seguir el enfoque de la mayoría de estudios modernos y la definimos como una serie de rasgos o características que posee cada individuo y que lo diferencia de los demás. [2] [3] [4]

Nuestra personalidad tiene un gran impacto en nuestras vidas. De ella van a depender nuestras decisiones más importantes y va a afectar tanto a nuestro estado de salud como a la determinación de muchas de nuestras preferencias. [5] Es por eso motivo que conociendo la personalidad de un usuario se puede tener información sobre sus gustos, preferencias, motivaciones, etc. Es decir, tener datos sobre su personalidad nos ayudará a intentar prever la motivación para actuar de una manera determinada u otra. [2] [3] [4] Aunque no podemos predecir el comportamiento exacto de una persona en una situación concreta sí nos permite predecir el comportamiento en lineas generales. [6]

La detección automática de los rasgos de la personalidad de una persona tiene muchas aplicaciones prácticas importantes que explicamos a continuación. [5]

Es útil en los siguientes ámbitos: En gestión de recursos humanos, para seleccionar candidatos según su personalidad, ya que pueden indicar la posible idoneidad así como el rendimiento en el trabajo y la satisfacción del trabajador. [3] [5]

En el contexto de análisis de sentimientos, usuarios con personalidad similar pueden tener gustos similares en cuanto a la compra de productos por internet o a la contratación de servicios. Por lo tanto es una herramienta útil para realizar recomendaciones de servicios y productos. [5].

Por otro lado, el análisis de mensajes de texto para la obtención de la personalidad incrementa la probabilidad de conseguir una relación satisfactoria en la búsqueda de pareja o de amistad en los servicios web de citas. [2]

Conocer la personalidad de un individuo nos va a ayudar a conocer mejor sus necesidades y nos va a permitir adaptar los sistemas de aprendizaje a distancia y del cuidado de la salud, ya que estos pueden adaptarse a las características del alumno y pacientes. [2] [4]

En las redes sociales, la búsqueda de indicios que marcan la personalidad del usuario puede ser útil para personalizar los mensajes que recibe el usuario. El tipo de personalidad está relacionado con los amigos que elegimos en las redes y con los gustos musicales.

También puede ser usado para predecir el voto en unas elecciones, para determinar la elección de una carrera profesional o las preferencias en las marcas comerciales entre otras cosas. [3]

El análisis de la personalidad en conversaciones puede ser útil en la identificación de lideres en la lucha contra el terrorismo. [2]

Hemos citado muchas de las múltiples aplicaciones aunque hay un abanico mucho más amplio.

Para obtener la personalidad vamos a utilizar un test de personalidad. Estos nos sirven para poder obtener los rasgos de conducta de la persona.

Los test analizan patrones de comportamiento humano ante determinadas situaciones de la vida diaria, para obtener las capacidades del individuo y su personalidad.

Los test de personalidad se realizan a un grupo de personas que acceden a realizarlo de manera voluntaria. Los usuarios tienen que responder a una serie de preguntas eligiendo la opción más adecuada a cada pregunta dentro de un conjunto limitado de preguntas. Estos test se pueden corregir con facilidad y los resultados se analizan estadísticamente. [7]

Dentro de la variedad de test que existen uno de los más importantes es el Eysenck Personality Questionnaire - Revised (EPQ-R) . Este test consiste en 94 preguntas con respuestas de Sí o No sobre los pensamientos y el comportamiento de una persona. Este test mide la Extraversión, el Neuroticismo, el Psicoticismo y la Sinceridad. [8] Para nuestro trabajo elegimos este test como primera opción pero durante el periodo de pruebas del bot los usuarios comentaron que les resultaba demasiado largo y pesado en el contexto de la aplicación.

Por ello, para que al usuario le resultase más ameno y dinámico la realización del test finalmente optamos por usar el test Overall PERsonality Assessment Scale (OPERAS) . Este test fue desarrollado por un grupo de psicólogos con el objetivo de obtener puntuaciones fiables en el modelo de personalidad Big Five, mediante un cuestionario más corto de 40 preguntas. El modelo Big Five de la personalidad descompone la personalidad en cinco rasgos: Extraversión, Estabilidad Emocional, Responsabilidad, Cordialidad y Apertura a la Experiencia. [9]

Cada rasgo se corresponde grosso modo con las siguientes características: [1]

- Extraversión: individuo sociable, jovial y muy activo.
- Estabilidad Emocional: tranquilidad y autocontrol.
- Responsabilidad: individuo consciente, trabajador y responsable.
- Cordialidad: individuo confiado y bondadoso.
- Apertura a la Experiencia: individuo creativo e imaginativo.

Nos encontramos diferentes estudios [2] [3] [4] [5] que buscan obtener la personalidad de manera indirecta. Es decir, en vez de usar tests, analizan distintos aspectos del comportamiento del usuario. Estos estudios se centran principalmente en el texto escrito y en el comportamiento social de los usuarios.

En [2] analizan conversaciones y pequeños textos escritos por usuarios para obtener la personalidad en el modelo Big Five. Para ello cuentan la frecuencia de aparición de cada una de las categorías del Linguistic Inquiry and Word Count (LIWC) (Investigación lingüística y conteo de palabras, en inglés) del texto del usuario y emplean diversos modelos de Aprendizaje Automático para predecir la personalidad. El LIWC es una base de datos que relaciona palabras con categorías psicológicamente significativas. [10] En este estudio para cada característica del test psicológico dividen al conjunto de usuarios en dos categorías, con igual número de sujetos y consiguen una precisión mayor al 55 % en casi todas las características psicológicas, frente al 50 % de precisión de un clasificador aleatorio.

En [3] analizan el texto escrito por usuarios de Twitter con una técnica similar y además añaden características propias de la red social como número de seguidores, número de personas a las que sigue el usuario y frecuencia de uso de la red social.

En [4] analizan diversas atributos de uso de la red social Facebook, como número de amigos, número de publicaciones y tiempo que lleva el usuario en Facebook para predecir la personalidad. En este estudio consiguen una precisión por encima del 60 % al dividir las características en 5 clases (muy bajo, bajo, medio, alto y muy alto) y una precisión por encima del 70 % al dividir las características en 3 clases (bajo, medio y alto).

En [5] usan redes neuronales convolucionales, un tipo de redes neuronales profundas, para obtener la personalidad a través de texto escrito. En este trabajo se obtiene una precisión por encima del 55 % en características binarias (por ejemplo, el usuario es o no extrovertido).

Por ello, se han realizado multitud de estudios para deducir la personalidad a partir del texto escrito [2] y el uso de redes sociales como Twitter [3] o Facebook [4]. Muchos de estos estudios han encontrado resultados significativos y tienen en común el uso de Aprendizaje Automático y el análisis del uso de varias categorías de palabras a través de bases de datos como el LIWC [10]. Aunque no hemos encontrado investigaciones para deducir la personalidad a partir del uso de bots conversacionales,

nuestro objetivo ha sido intentar abrir ese campo de investigación.

Por ello este trabajo busca solventar en parte esta situación y crear una infraestructura a partir de la cual se puedan llevar a cabo nuevos estudios.

Para ello hemos creado un bot conversacional que administra el test psicológico OPERAS y realiza una serie de preguntas sobre los gustos televisivos con el fin de relacionar tanto el uso del bot como el gusto televisivo con la personalidad.

El desarrollo de este bot nos ha supuesto técnicamente un reto debido a la gran cantidad de tecnologías distintas utilizadas durante el desarrollo de este bot y el gran campo de investigación que se puede llegar a abrir.

Para la recolección de datos hemos promocionado a los usuarios un bot conversacional por la aplicación de mensajería Telegram. Para ello se distribuyó un enlace por Telegram para poder acceder al bot entre conocidos, pidiéndoles que lo difundieran lo más ampliamente posible.

Este trabajo se estructura en seís capítulos:

1.– Introducción: donde se explica la motivación del trabajo, los distintos modelos de personalidad, los objetivos y la estructura.

2.– Estado del arte: donde se explican las distintas tecnologías para construir un bot. Además se realiza un resumen de trabajos anteriores similares.

3.– Análisis de Requisitos: donde se enumeran los requisitos funcionales y no funcionales.

4.– Diseño e implementación: donde se realiza un resumen de los elementos del framework Dialogflow y como se usan en la arquitectura general del programa y el diagrama de flujo resumido de la interacción del usuario. Además de la arquitectura general de análisis.

5.– Resultados: donde se explican las características, las relaciones más significativas y los resultados del Aprendizaje Automático.

6.– Conclusiones y trabajo futuro: que se ha aprendido y que se podría hacer en trabajos posteriores.

2

Estado del Arte

Los bots conversacionales despiertan hoy en día un gran interés comercial, como ejemplo tenemos los asistentes personales basados en voz como Cortana, Siri y Alexa.

En el mundo académico se llevan estudiando desde los años 60(bot psicólogo ELIZA creado por el Massachusetts Institute of Technology (MIT) en 1964). Este bot era capaz de identificar palabras y un contexto mínimo pero le faltaba la capacidad de mantener la conversación [11].

Posteriormente, también en el ámbito académico surge el bot conversacional Artificial Linguistic Internet Computer Entity (A.L.I.C.E.) . Este bot esta construido usando el lenguaje Artificial Inteligence Markup Language (AIML) , un lenguaje basado en eXtensible Markup Language (XML) y usado en la mayoría de bots conversacionales actuales. [11]

Vamos a estudiar primero la clasificación de los bots conversacionales y después pasaremos a resumir las tecnologías más usadas en la actualidad.

En cuanto a la clasificación los bots se pueden clasificar según dos criterios: [11]

- El enfoque usado para diseñarlos
- Las capacidades del bot.

Actualmente hay tres enfoques de diseño [11]:

- El diseño basado en reglas: hay un conjunto de reglas muy definidas y respuestas asociadas. Este diseño resulta muy rígido y es el enfoque más primitivo de todos los bots conversacionales, hoy en día en desuso.
- El diseño basado en recuperación: usan respuestas predefinidas y conjuntos de ejemplos para entrenar modelos de Aprendizaje Automático. Estos modelos suelen ser específicos del campo de Natural Language Processing (NLP) . Tienen como conceptos centrales la intención del usuario y el contexto de la conversación, es decir, tienen en cuenta los mensajes que se han escrito antes. Se adaptan mejor a las variaciones al plantear una pregunta, resultando más dinámico. Este es el que hemos elegido para nuestro trabajo.

- El diseño basado en generación: no tiene respuestas predefinidas. Busca imitar una conversación sin restricciones convirtiéndose en el más dinámico. Procesan grandes cantidades de texto con el objetivo de aprender a conversar, aunque actualmente está todavía en investigación.

En cuanto a las capacidades hay dos tipos: [11]

- Los bots orientados a una tarea: en esta categoría se incluyen casi todos los bots actuales. Se utilizan los diseños basados en reglas y los basados en recuperación. Entre estos podemos destacar Cortana, Siri y Alexa.
- Los bots no orientados a una tarea: estos bots se diseñan con el objetivo de mantener una conversación sobre cualquier tema y su diseño puede ser basado en recuperación o basado en generación. Estos no han tenido mucho éxito. Un ejemplo de este tipo de bot es Tay, un bot de Microsoft para Twitter lanzado en 2016 y desconectado a las pocas horas debido a que producía mensajes ofensivos.

En cuanto al punto de vista comercial, la mayoría de aplicaciones de mensajería (Telegram, Facebook Messenger, Skype) han creado APIs (Aplication Programming Interface) para posibilitar la creación e integración de bots. [12]

Además, las grandes empresas tecnológicas han creado asistentes virtuales como Cortana, Siri, Google Now y Alexa creadas por Microsoft, Apple, Google y Amazon respectivamente. Estos asistentes virtuales son un tipo de bot conversacional que busca la realización de tareas sencillas como programar una alarma en el teléfono o mandar un mensaje y la respuesta a preguntas sencillas como ¿Que tiempo hace hoy?.

En cuanto a la tecnología comercial para la construcción de bots tenemos tres grandes alternativas:

- Azure Bot Service de Microsoft.
- Amazon Lex de Amazon.
- Dialogflow de Google.

Cada una de las alternativas está integrada en los servicios de computación en la nube de la compañía que lo ha desarrollado (Microsoft Azure, Amazon Web Service y Google Cloud). Todas estas alternativas ofrecen integración con las aplicaciones de chat más populares. [12] [13] [14]

Azure Bot Service es un servicio en la nube que junto con el Bot Framework (el framework de microsoft para construir los bots) permite la creación de bots en C# y JavaScript. Incluye módulos para bots QnA (pregunta-respuesta) y para procesado de lenguaje Natural (módulo LUIS). [15]

Amazon Lex es el servicio en la nube de Amazon para la creación de bots, la mayor parte de

funciones se pueden especificar desde una interfaz gráfica. Se puede integrar con AWS Lambda para la lógica de negocio necesaria. [16]

Dialogflow es el servicio en la nube de Google para crear bots. Es el sistema más sencillo, permite la creación de los bots desde una interfaz gráfica y permite programar la lógica de negocio en un servidor NodeJS integrado o en cualquier servidor externo (webhook). [17] [18] [19].

Para la construcción de nuestro bot hemos elegido Dialogflow debido a su facilidad de implementación y flexibilidad. Además, nos permite poder disponer de la capacidad de controlar cómo se almacenan los datos necesarios para la investigación al implementar parte de la lógica en nuestro servidor. Dialogflow implementa bots basados en recuperación y orientados a tareas y en su versión básica es gratuito.

Por otro lado, hemos elegido la plataforma de mensajería Telegram para su distribución por ser una plataforma popular que permite bots. La plataforma de mensajería Whatsapp, más popular que Telegram, se descartó debido a la falta de soporte de bots por parte de la misma.

3

Análisis de Requisitos

En esta sección enumeraremos los requisitos funcionales y no funcionales de la aplicación desarrollada en este trabajo.

Para la obtención de los requisitos se dispuso de la ayuda del tutor de este trabajo actuando como usuario del software a desarrollar. Se hizo así ya que los resultados de este trabajo pueden ayudar al tutor en futuros proyectos de investigación.

3.1. Requisitos Funcionales

Los requisitos funcionales describen las distintas funciones que debe tener la aplicación. En este caso son los siguientes:

RF-1.– Administrar un test psicológico y dar el resultado al usuario.

RF-2.– Funcionar a través de una plataforma de mensajería

RF-3.– Recolectar y analizar la conversación del usuario elaborando un perfil.

RF-4.– Debe intentar reconducir al usuario cuando no entienda la pregunta o se equivoque en la respuesta.

RF-5.– Obtener un modelo de Aprendizaje Automático para predecir la personalidad a través del perfil del usuario.

RF-6.– Recomendar series de televisión de distintas plataformas de streaming.

3.2. Requisitos No Funcionales

Los requisitos no funcionales son aquellos que describen otros aspectos del diseño de la aplicación. En este caso son los siguientes:

RNF-1.– Ser fácil de usar por el usuario.

RNF-2.– Ser accesible a una gran parte de la población.

RNF-3.– Ser interesante para el usuario.

RNF-4.– Tener un bajo coste de desarrollo.

RNF-5.– Debe poder funcionar en la mayoría de móviles actuales.

4

Diseño e implementación

En este apartado vamos a hablar del diseño e implementación del bot conversacional y del sistema para el análisis de los resultados.

4.1. Bot Conversacional

En cuanto al diseño del bot se pueden ver cuatro componentes principales tal como se muestra en la Figura 4.1:

- Telegram: la aplicación de mensajería instantánea sobre la que funciona el bot. Este componente es una aplicación movil ya desarrollada.
- Dialogflow: el framework sobre el que construimos el bot. El desarrollo sobre este framework se realiza mediante una interfaz web sin necesidad de programar.
- Python Webhook: se encarga tanto de registrar de cada mensaje el texto original y los parámetros más importantes como de construir las respuestas personalizadas. Este componente es un servidor web desarrollado en Python usando el framework Flask. Además contiene código SQL para la realización de las consultas a PostreSQL.
- PostgreSQL: la base de datos donde se guarda toda la información de interés. Este componente es una base de datos ya desarrollada. Lo único necesario es definir el esquema de la base de datos.

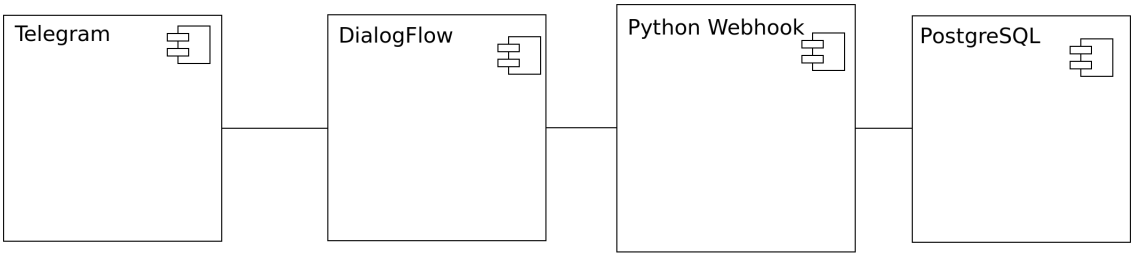


Figura 4.1: Componentes principales

El flujo de control de la aplicación es el siguiente:

1.– El usuario escribe un mensaje a través de Telegram

2.– Telegram envia el mensaje a Dialogflow

3.– Dialogflow analiza el mensaje y obtiene la intención del usuario y extrae los parámetros del mensaje.

4.– Dialogflow envía el mensaje y el análisis al Python WebHook en formato JSON

5.– El Python WebHook envía a PostgreSQL las consultas SQL necesarias para almacenar los datos recibidos.

6.– El Python WebHook si es necesario calcula la respuesta personalizada y se la envía a Dialogflow. Si no es necesaria la respuesta personalizada se envía un JSON vacio.

7.– Dialogflow envía a Telegram la respuesta personalizada o ,si esta no existe, la respuesta por defecto.

En cuanto al flujo de conversación del bot, el bot empieza saludando con el siguiente mensaje cuando el usuario abre el chat de Telegram(cuadro 4.1).

Hola, soy un bot conversacional diseñado para recomendar series de tv. Para ello vamos a realizar un test de personalidad y algunas preguntas sobre tus preferencias. Responde Ok para empezar.

Cuadro 4.1: Mensaje de bienvenida

Después le pregunta la edad y el sexo y le da una pequeña explicación sobre como realizar el test psicológico OPERAS (cuadro 4.2).

Una vez que el usuario ha respondido a todas las preguntas, le presenta los resultados y le sugiere empezar el segundo test(cuadro 4.3).

A continuación se presentan un conjunto de frases en relación con tu forma de pensar y de actuar. Has de decidir hasta qué punto te describen cada una de las afirmaciones. No hay respuestas correctas ni incorrectas, ni tampoco respuestas buenas o malas. Las alternativas de respuesta para cada afirmación son:
1. Completamente en desacuerdo
2. Bastante en desacuerdo
3. Ni de acuerdo ni en desacuerdo
4. Bastante de acuerdo
5. Completamente de acuerdo
Responde numéricamente. Para empezar di Ok.

Cuadro 4.2: Explicación del test OPERAS

Hemos terminado con el test psicológico. Los resultados son:
-Extroversión 22 puntos
-Estabilidad emocional 53 puntos
-Responsabilidad 50 puntos
-Cordialidad 51 puntos
-Apertura a la experiencia 40 puntos
Esto te situa en los siguientes percentiles:
-0 % de Extroversión
-60 % de Estabilidad emocional
-51 % de Responsabilidad
-54 % de Cordialidad
-16 % de Apertura a la experiencia
Has finalizado el test. Ahora puedes pedir la recomendación de una serie.Si quieres comenzar di Ok.

Cuadro 4.3: Resultados del test

En el segundo test se le realiza al usuario una serie de preguntas sobre sus gustos en cuanto a series de televisión, con el objetivo de recolectar datos(cuadro 4.4).

Empezamos el test para recomendar una serie. Separa por comas cuando escribas más de una respuesta.¿Qué plataforma de Streaming usas?

Cuadro 4.4: Explicación del segunto test

Al finalizar las preguntas, el bot le recomienda una serie de televisión en base a algunas de las respuestas que el usuario ha dado (la recomendación no tiene todas las respuestas en cuenta puesto que es una característica secundaria)(cuadro 4.5).

Ya hemos terminado y mi recomendación es Better Things.Espero que te guste

Cuadro 4.5: Recomendación de un serie

Como pudimos observar que muchos usuarios agradecían al bot la recomendación y se despedían de él, añadimos una respuesta de cortesía(cuadro 4.6 y 4.7 respectivamente).

De nada

Cuadro 4.6: Mensaje de Cortesia

Adios

Cuadro 4.7: Mensaje de Despedida

Ahora vamos a ver cada uno de los elementos del diseño con mayor detalle.

4.1.1. Dialogflow

Permite la integración de manera relativamente sencilla con las aplicaciones de mensajería, ya que solo hay que registrar el bot en la aplicación de mensajería e introducir las claves en Dialogflow (el proceso exacto depende de la aplicación).

Además, permite aumentar la funcionalidad del bot mediante Webhooks, que pueden tanto alterar el comportamiento del bot como realizar cualquier acción externa necesaria.

Los elementos clave de Dialogflow son: [20]

- Contexts (Contextos): Los contextos almacenan información sobre la conversación, tales como las intenciones anteriores o los parámetros de un mensaje. Un tipo especial de contexto son los "followup", que se nombran como la intención a la que preceden + "-followup" y sirven para dirigir el flujo de la conversación. Tienen una vida limitada.
- Entities (Entidades): Sirven para definir categorías de palabras, como por ejemplo colores. Permiten definir un conjunto de valores posibles, junto con sinónimos y coincidencias aproximadas, para reconocer una palabra aunque esté mal escrita, así como ser definidas mediante expresiones regulares.
- Intents (Intenciones): Marcan la intención de un usuario al mandar un mensaje. Cada intención se define según los contextos de entrada (los contextos que tienen que estar activos para seleccionar la intención) y unos mensajes de entrada de ejemplo que pueden estar parametrizados. Además se definen las respuestas, los contextos de salida y si se llama o no al Webhook para modificar lo anterior. Existen intenciones especiales llamadas "-fallback", para cuando no reconoce ninguno de los textos y se seleccionan únicamente según el contexto.

Para cada mensaje que manda el usuario, así como para eventos creados por las aplicaciones de mensajería (ej: TELEGRAM_WELCOME) se asocia una intención y se manda una petición al Webhook con el texto, los contextos, la intención y los parámetros del mensaje.

Entonces, el Webhook decide si responder un JavaScript Object Notation (JSON) vacío o mandar el texto del mensaje que se mandará al usuario. Si manda un JSON vacío, Dialogflow selecciona el texto que tiene predefinido, sino manda el definido por el Webhook.

Ahora explicaremos como se crea y configura un intent. Una vez que se ha iniciado sesión en Dialogflow aparece la pantalla de la figura 4.2 con la lista de intents. Para crear un intent se pulsa sobre "CREATE INTENT".Entonces aparece la pantalla de la figura 4.3.

Lo primero que se hace es elegir un nombre para el intent.

Después en la sección "contexts" figura 4.4 se añaden los contextos de entrada y de salida introduciendo el nombre del contexto que se crea en el momento.

A continuación en la sección "Training phrases" se definen ejemplos de las frases que puede introducir un usuario, con parametros figura 4.6 o sin ellos figura 4.5.

Después en la sección "Responses" se define las posibles respuestas predefinidas (Si se eligen varias Dialogflow coge una al azar). figura 4.7

Por último se activan las dos opciones de la sección "Fulfillment" figura 4.8 para que al reconocer ese intent llame al WebHook y para que acepte la respuesta que le envía el WebHook en vez de dar la respuesta predefinida.

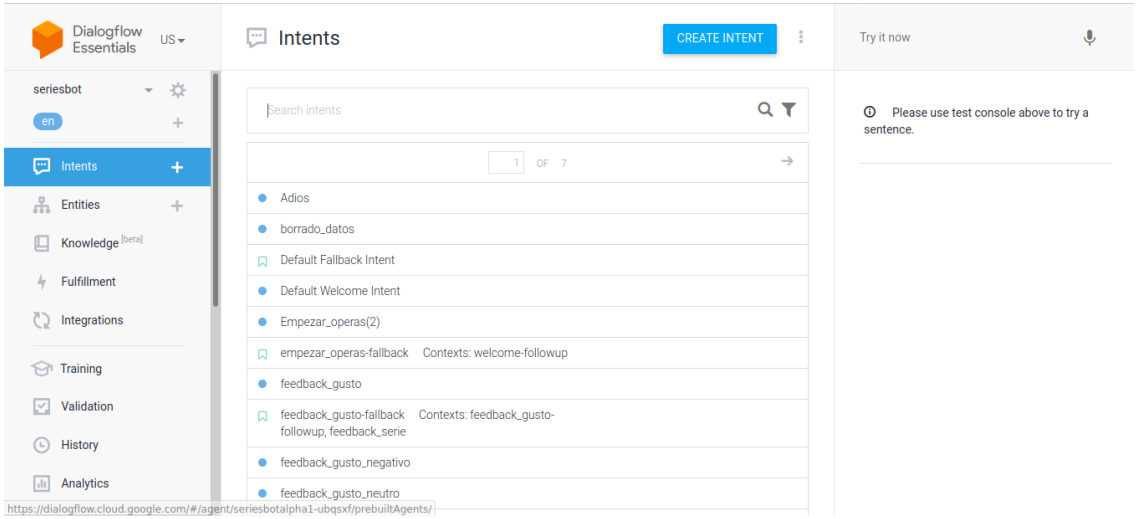


Figura 4.2: lista de Intents de Dialogflow

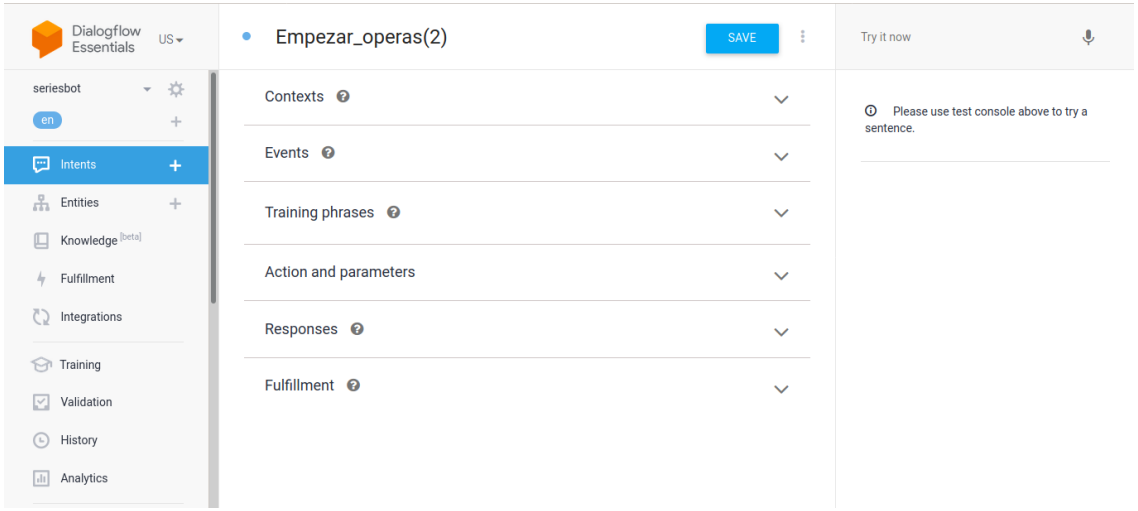


Figura 4.3: Secciones de un Intent

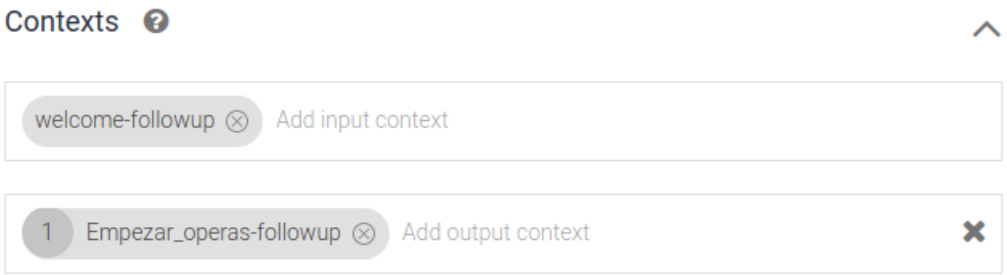


Figura 4.4: Contexts de un Intent de Dialogflow

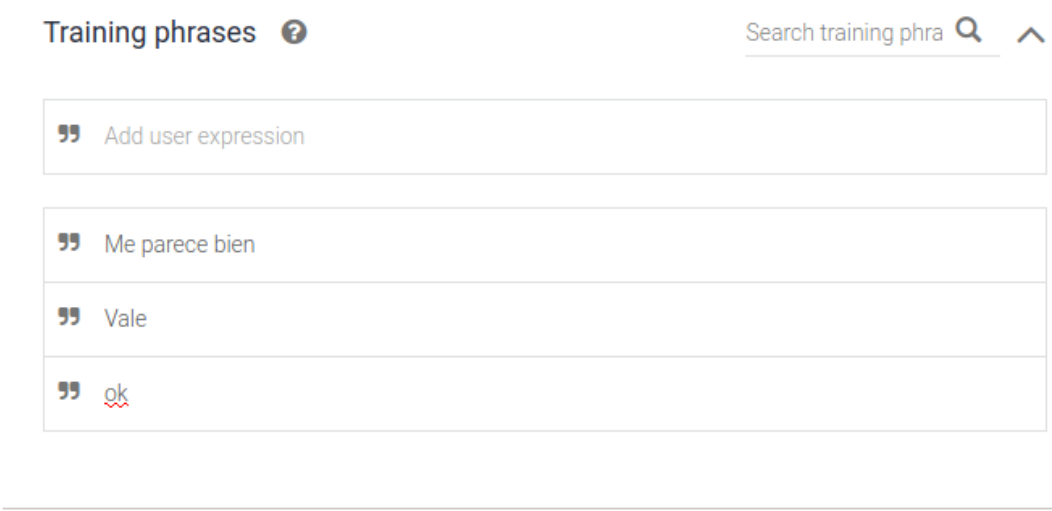


Figura 4.5: Frases de entrenamiento de un Intent de Dialogflow

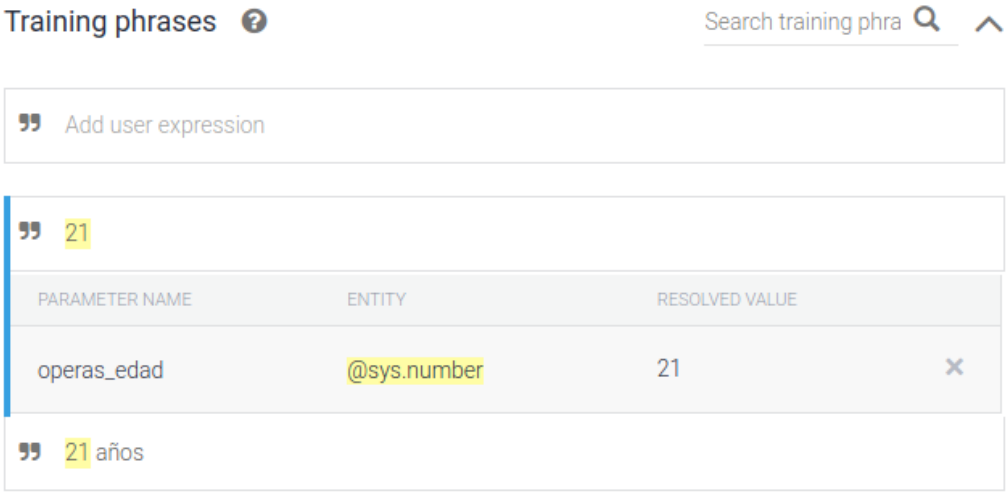


Figura 4.6: Frases de entrenamiento de un Intent de Dialogflow con parámetros

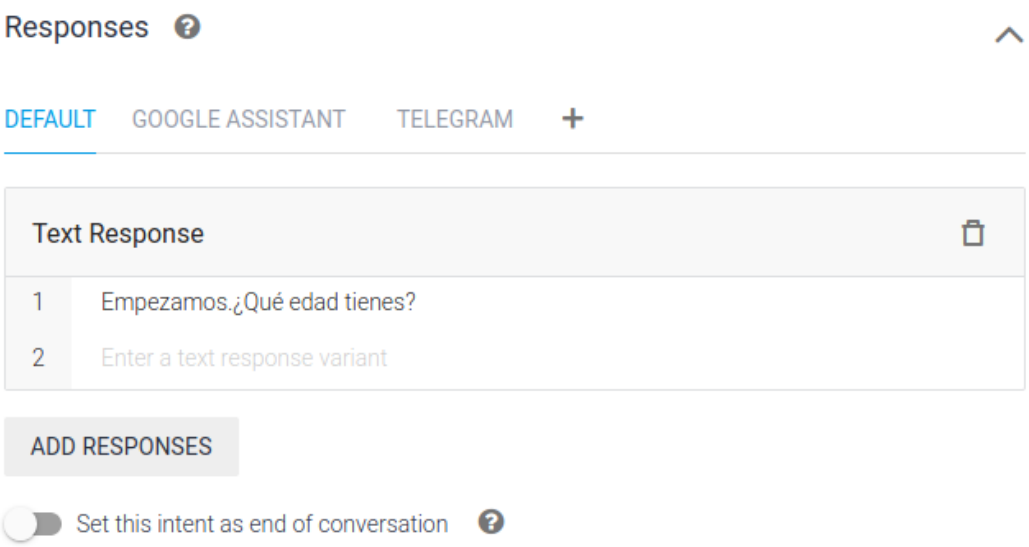


Figura 4.7: Respuestas predefinidas de un Intent de Dialogflow

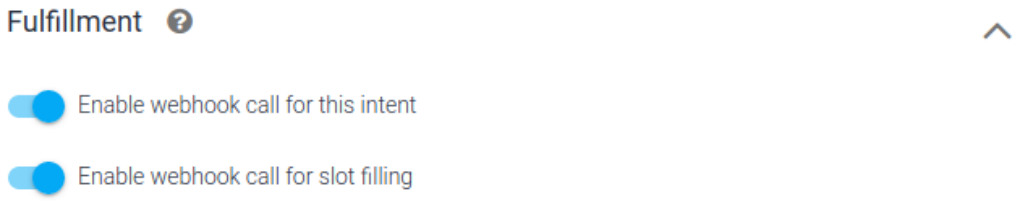


Figura 4.8: Parametros de configuración de un Intent de Dialogflow respecto al WebHook

4.1.2. Python Webhook

El Webhook para este bot conversacional se ha programado en Python y consiste en un aplicación web Flask que recibe peticiones y devuelve respuestas en formato JSON a tráves del protocolo HTTPS. Como servidor web WSGI se usa Gunicorn.

Esta parte del sistema se divide en cuatro módulos:

- __init__.py: módulo principal donde se implementa la lógica del bot conversacional.
- database.py: se define la interfaz entre la aplicación y la base de datos. Este módulo usa la librería SQLAlchemy.
- analitics.py: en este módulo se calculan las estadísticas de uso, así como datos lingüísticos, el número de errores ortográficos o el número de palabras de cada categoría del LIWC .
- operas.py: en este módulo se calculan las puntuaciones del test OPERAS .

4.1.3. Base de Datos

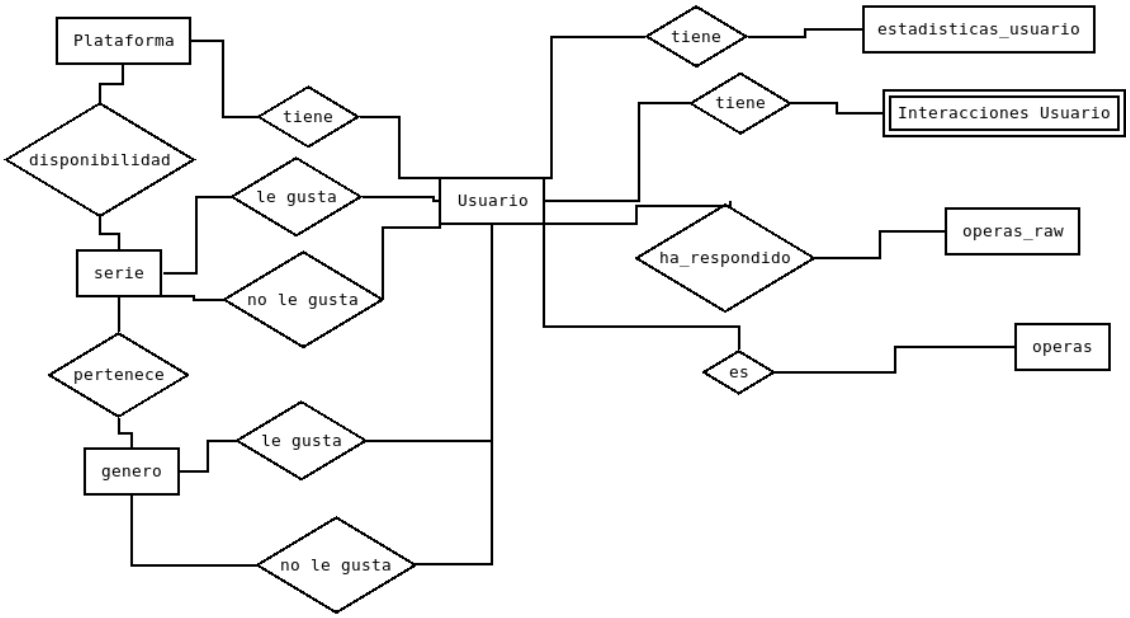


Figura 4.9: Esquema E-R simplificado

Para la base de datos, hemos elegido el gestor de bases de datos PostgreSQL, debido a su alto rendimiento y que al ser el gestor de bases de datos más usado en la EPS (Escuela Politécnica Superior), ya está instalado en los servidores y eso nos permite instalar el proyecto sin hacer grandes cambios en los sistemas informáticos. No se considero usar sistemas NoSQL debido a que por el tamaño reducido y la estructura de los datos no presentaban ventajas significativas.

En el esquema E-R simplificado de la base de datos(figura 4.9) se pueden diferenciar dos partes:

Una primera parte correspondiente al test psicológico y al funcionamiento básico del bot con las entidades: usuario, estadísticas_usuario, operas_raw (con las respuestas del test psicológico) y operas (con los resultados del test psicológico).

Y una segunda parte, correspondiente a la recomendación de series de televisión con las entidades: plataforma (plataformas de streaming como Netflix o HBO), serie y género.

Este esquema una vez desarrollado forma 16 tablas:

- genre (género): define cada uno de los géneros de series de televisión.
- genre_sinonimo (sinónimos de género): define nombres alternativos para los géneros de series de televisión.
- genre_user_dislike: los géneros que el usuario ha dicho que no le gustan.
- genre_user_like: los géneros que el usuario ha dicho que le gustan.
- operas: resultados de los test psicológicos.
- operas_raw: respuestas a las preguntas del test psicológico.
- platform: plataformas de streaming definidas.
- serie: lista de series de televisión para el sistema de recomendación.
- serie_platform: plataformas de streaming en las que aparece cada serie.
- serie_user_like: las series que el usuario ha dicho que le gustan.
- serie_user_dislike: las series que el usuario ha dicho que no le gustan.
- user_stats : estadísticas del usuario, como número de mensajes enviados, número de palabras usadas, número de palabras de las categorías del LIWC .
- users: usuario con el identificador en Dialogflow y las repuestas a preguntas que no corresponden a ninguna otra tabla.
- users_interactions: lista de mensajes mandados por los usuarios.
- users_platform: plataforma a las que esta suscrito cada usuario.
- users_recomendation: recomendaciones que ha realizado el sistema.

4.2. Sistema de análisis de resultados

Para analizar los resultados del estudio se han desarrollado una serie de módulos en Python.

- features.py: extrae las características y devuelve una lista de características con el nombre de cada característica y pares id_usuario-valor.
- distribucion1variable.py: sirve para analizar cada variable por separado.

- correlacion2variables.py: sirve para analizar las correlaciones entre las variables psicológicas y las variables de uso del bot.
- machine_learning.py: sirve para usar las herramientas disponibles en sklearn para encontrar un modelo para cada variable usando métodos de Regresión.
- machine_learning_clasificacion.py: cumple la misma función que machine_learning.py salvo que usa métodos de clasificación en vez de usar metodos de Regresión

Los atributos de features.py consisten en:

Caracteristicas de uso del bot:

- n_mensajes: número de mensajes enviados.
- n_series: número de series mencionadas.
- n_series_like: número de series que le han gustado al usuario.
- n_series_dislike: número de series que no le han gustado al usuario.
- n_generos: número de géneros mencionados.
- n_generos_like: número de géneros que le han gustado al usuario.
- n_generos_dislike: número de géneros que no le han gustado al usuario.
- numero_conversaciones: número estimado de conversaciones que el usuario ha tenido con el bot.
- tiempo_entre_mensajes: tiempo medio entre mensajes de una misma conversación.
- longitud_media_mensajes: longitud media de cada mensaje.
- miedo(corpus_sentimientos): número de palabras relacionadas con el miedo. [21]
- liwc1...liwc68 : número de menciones de cada categoría del LIWC . [22]
- mencionado-género: 0 si no ha mencionado el género, 1 si lo ha mencionado.
- gusto-género: -1 si no le gusta el género, 0 si no lo ha mencionado y 1 si le gusta.
- respuesta longitud serie: -1 si le gustan series cortas, 0 si es indiferente y 1 si le gustan series largas.
- respuesta longitud capitulo: -1 si le gustan series con capítulos cortos, 0 si es indiferente y 1 le gustan series con capítulos largos.
- respuesta_espanola: -1 si el usuario prefiere series extranjeras, 0 si es indiferente y 1 si prefiere series españolas.
- respuesta_actores_conocidos: -1 si el usuario prefiere series sin actores conocidos, 0 si es indiferente y 1 si el usuario prefiere series con actores conocidos.
- respuesta_version_original: -1 si el usuario prefiere series dobladas, 0 si es indiferente y 1 si el usuario prefiere series en versión original.

- respuesta_terminar_serie: -1 si el usuario tiende a abandonar una serie, 0 si es indiferente y 1 si el usuario tiende a terminar una serie.
- respuesta_hechos_reales: -1 si el usuario prefiere series de ficción, 0 si le es indiferente y 1 si prefiere series basadas en hechos reales.
- respuesta_solo_o_acompanado: -1 el usuario prefiere ver las series acompañado, 0 si le es indiferente y 1 si prefiere ver las series solo.
- respuesta_momento_ver_serie: -1 si el usuario ve las series el fin de semana, 1 si ve las series entre semana y 0 si le es indiferente.
- respuesta_premiadas : -1 si prefiere series no premiadas, 0 si le es indiferente y 1 si prefiere series premiadas.
- respuesta_uno_o_varios_capitulos: -1 si el usuario ve varios capítulos, 0 si le es indiferente y 1 si ve solo un capitulo
- mensajes_no_entendidos: número de mensajes en los que el bot no entendió el mensaje del usuario.
- hora_de_conexión: hora en la que se mando el primer mensaje, codificada como el número de minutos pasados desde medianoche.
- n_mayusculas: número de mayúsculas empleadas por el usuario.
- n_signos_de_exclamación: número de signos de exclamación empleados por el usuario
- n_errores: número de errores ortográficos estimados.
- prop_mayusculas: número de mayúsculas dividido entre el número de caracteres.
- prop_errores: número de errores dividido entre el número de mensajes.

Características del usuario

- Extroversión (EX): puntuación para extroversión en el test OPERAS
- Estabilidad Emocional (EE): puntuación para Estabilidad Emocional en el test OPERAS
- Responsabilidad (CO): puntuación para Responsabilidad en el test OPERAS
- Cordialidad (AG): puntuación para Cordialidad en el test OPERAS
- Apertura a la experiencia (OP): puntuación para Apertura a la experiencia en el test OPERAS
- Edad: edad del usuario

En el módulo de Aprendizaje Automático machine_learning.py, se intenta encontrar un modelo de Regresión para predecir las características del usuario a partir de las características de uso del bot.

Para ello, se usa la librería de Aprendizaje Automático sklearn y se prueban de manera automática varios modelos, dividiendo el modelo en tres fases mediante la clase Pipeline de sklearn.

Estas fases son:

- re-escalado de características
- selección de variables
- algoritmo de Aprendizaje Automático.

En cuanto a re-escalado de características probamos:

- Standard Scaler: normaliza las características restando la media y dividiendo entre la varianza.
- Power Transformer: Aproxima las características a una distribución normal
- Sin re-escalado

En cuanto a selección de variables probamos:

- SelectKBest con 5,10,20,40 y 80 con los criterios f_regression (correlación entre las características de uso y las características del usuario) y mutual_info_regression (correlación entre distintas características de uso).
- Variance Threshold con umbral 0.1,0.2 y 0.3 (Elimina las características con menor varianza)
- Sin selección de variables

En cuanto a algoritmos de Aprendizaje Automático probamos:

- modelos lineales (LinearRegression,ElasticNet,BayesianRidge,SDGRegressor),
- modelos basados en vecinos próximos (KNeighborsRegressor),
- modelos basados en árboles de decisión (DecissionTreeRegressor),
- modelos basados en Support Vector Machine (SVR),
- redes neuronales (MLPRegressor)
- modelos de tipo ensemble (GradientBoostingRegressor, RandomForestRegressor)

Para seleccionar el mejor modelo, calculamos las métricas r2_score (correlación entre valor real y valor predicho) y el mean_absolute_error (error absoluto medio) además de usar como referencia un modelo base que devuelve siempre la media.

En el módulo machine_learning_clasificacion.py se intenta encontrar un modelo para clasificar al usuario en tres categorías por cada característica del usuario. Estas categorías son: bajo, medio y alto.

Al igual que en el módulo anterior se divide el modelo en tres fases mediante la clase Pipeline de sklearn. La fase de re-escalado de características y de selección de variables es común y cambian la fase del algoritmo de Aprendizaje Automático y la métrica de evaluación.

En este módulo se prueban:

- modelos lineales(RidgeClassifier, LogisticRegression, SDGClassifier)
- modelos basados en vecinos próximos(KNeighborsClassifier)
- modelos basados en árboles de decisión(DecisionTreeClassifier)
- modelos basados en Support Vector Machine (SVC)
- redes neuronales (MLPClassifier)
- modelos basados en probabilidad (CategoricalNB)
- modelos tipo ensemble (RandomForestClassifier)

En la siguiente sección, mostraremos estadísticas sobre los parámetros y los resultados del aprendizaje automático

5

Análisis de Resultados

En esta sección analizaremos los resultados obtenidos al analizar los datos.

Durante los 24 días de la realización del estudio, han usado el bot con éxito 433 personas.

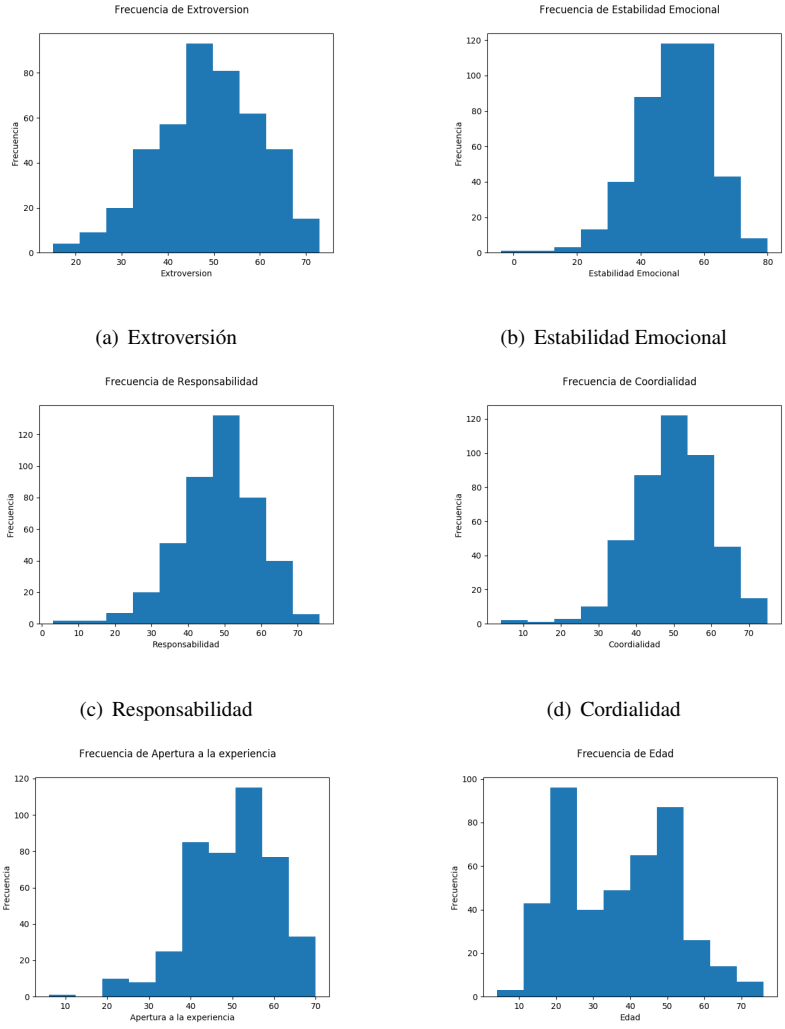
Primero vemos las gráficas de las variables a predecir(figura 5.1) y una tabla con los parámetros estadísticos obtenidos.

Como se puede observar en la figura 5.1, la mayoría de usuarios obtienen puntuaciones medias y unos pocos obtienen puntuaciones más lejanas a la media.

En la tabla 5.1, se observa que las distribuciones tienen una media cercana a 50 y una desviación típica cercana a 10, que corresponde a las especificaciones del test psicológico, aunque no coincide con precisión. De todas formas, es lo bastante parecido a las especificaciones del test OPERAS como para sugerir que la implementación desde el punto de vista de la ingeniería informática es correcta y que se necesitan más estudios para saber si la metodología causa algún sesgo.

En cuanto a la correlación entre el uso del bot y las puntuaciones en el test psicológico, se observa que la mayoría de correlaciones son muy débiles, pero hay algunas significativas son mostradas en las tablas 5.2 y 5.3 y se explican a continuación:.

- La Extraversión parece relacionarse negativamente con liwc29 (que corresponde con la categoría ***oir***. Ver Anexo A) y mencionar el género juvenil.
- La Estabilidad Emocional parece relacionarse con mencionar el género juvenil.
- La Responsabilidad parece relacionarse positivamente con preferir las series basadas en hechos reales y series españolas, con el gusto por el género policíaco así como con liwc49 (que corresponde con palabras relacionadas con ***trabajo***). También parece relacionarse negativamente con liwc15 (que corresponde con ***palabras optimistas***) y liwc50 (que corresponde con palabras relacionadas con ***logros***) y con el gusto por las series en versión original.
- La Cordialidad parece relacionarse negativamente con el número de mensajes no entendidos por el bot. Los mensajes no entendidos pueden deberse a un error ortográfico, a una palabra poco común, a una manera de responder no prevista al programar o que el usuario



(a) Extroversión

(b) Estabilidad Emocional

(c) Responsabilidad

(d) Cordialidad

(e) Apertura a la experiencia

(f) Edad

Figura 5.1: Distribuciones de frecuencia de cada variable a predecir

intente confundir al bot, entre otras razones.

- La Apertura a la Experiencia parece relacionarse positivamente con liwc16 (que corresponde a ***Emociones negativas***), liwc18 (que corresponde a palabras relacionadas con el ***enojo***), liwc30 (que corresponde a palabras relacionadas con el ***sentir***), el gusto por el drama y con ver solo un capitulo de una serie. Y también parece relacionarse negativamente con liwc43 (Palabras relacionadas con ***abajo***), liwc46 (palabras de la categoría ***Moción***), el gusto por las series de acción, la longitud de las series, la presencia de actores conocidos y los mensajes no entendidos por el bot.
- La edad se relaciona con muchas características pero las más significativas son de manera positiva: liwc49 (que corresponde con palabras relacionadas con ***trabajo***), el gusto por las series españolas, el gusto por las series basadas en hechos reales y ver un solo capitulo. Y de manera negativa las más significativos son: ver las series en versión original y ver las series acompañado.

	media	**desviación típica**
Extroversión	48.69	11.09
Estabilidad Emocional	50.56	11.53
Responsabilidad	48.4	10.99
Cordialidad	49.57	10.12
Apertura a la experiencia	49.91	10.2
Edad	37.01	14.94

Tabla 5.1: Parámetros estadísticos de las variables

	EX	**EE**	**CO**	**AG**	**OP**	**Edad**
liwc15	-0.07	-0.03	-0.14*	-0.06	0.01	-0.16*
liwc16	0.01	-0.00	0.08	-0.01	0.13*	-0.09
liwc18	0.05	0.00	0.03	0.01	0.15*	0.01
liwc29	-0.14*	-0.08	-0.06	-0.05	0.03	0.04
liwc30	-0.02	-0.05	-0.02	0.00	0.14*	-0.02
liwc37	0.01	-0.05	0.01	0.04	-0.01	-0.14*
liwc43	-0.00	0.07	-0.09	-0.03	-0.15*	-0.08
liwc46	0.02	0.09	0.00	0.07	-0.14*	-0.12
liwc49	0.05	0.04	0.15*	-0.01	0.11	0.22*
liwc50	0.00	0.02	-0.15*	-0.01	0.01	-0.19*
liwc54	-0.03	-0.01	-0.05	0.01	0.10	-0.14*

Tabla 5.2: Características LIWC. Características con al menos p<0.01 en una variable. Se remarcan pares de características con p<0.01

	EX	EE	CO	AG	OP	Edad
n_series	0.11	0.03	0.03	-0.05	0.11	-0.19*
n_series_like	0.11	0.02	0.02	-0.04	0.11	-0.20*
n_generos	-0.03	-0.07	-0.05	-0.04	0.04	-0.22*
n_generos_like	-0.05	-0.05	-0.05	-0.07	0.06	-0.22*
tiempo entre mensajes	0.06	0.06	0.02	-0.02	0.05	0.16*
longitud media mensajes	0.02	-0.00	0.02	0.01	0.12	-0.21*
mencionado-Comedia	-0.04	-0.04	-0.02	0.05	0.08	-0.17*
mencionado-Romance	0.02	-0.00	-0.05	-0.07	-0.05	-0.18*
mencionado-Policiaca	0.07	0.06	0.18*	-0.03	0.08	0.15*
gusto-Policiaca	0.07	0.04	0.19*	-0.03	0.09	0.17*
gusto-Drama	-0.05	-0.08	-0.08	-0.05	0.13*	-0.07
gusto-Terror	-0.08	0.02	-0.12	-0.09	-0.06	-0.21*
mencionado-Acción	-0.01	0.07	-0.01	0.04	-0.12	-0.19*
gusto-Acción	0.06	0.07	-0.01	0.03	-0.14*	-0.15*
gusto-Ciencia ficción	0.03	-0.07	-0.12	-0.11	-0.08	-0.14*
mencionado-Fantasia	-0.08	-0.08	-0.07	0.01	-0.01	-0.19*
gusto-Fantasia	-0.04	-0.05	-0.09	-0.00	0.02	-0.15*
mencionado-Juvenil	-0.13*	-0.16*	-0.04	-0.04	-0.03	-0.11
mencionado-Bélica	-0.05	-0.04	0.04	-0.01	0.03	0.13*
respuesta longitud de serie	-0.02	0.05	-0.08	-0.10	-0.18*	-0.22*
respuesta longitud capitulo	0.01	0.05	-0.02	-0.03	-0.08	-0.14*
respuesta espanola	0.02	0.02	0.13*	0.03	-0.00	0.23*
respuesta actores_conocidos	0.07	0.04	-0.02	-0.01	-0.13*	0.00
respuesta version original	-0.09	-0.07	-0.16*	-0.03	0.10	-0.27*
respuesta hechos reales	0.09	0.10	0.17*	0.11	0.10	0.22*
respuesta solo o acompanado	-0.05	-0.00	-0.05	-0.11	-0.01	-0.25*
respuesta uno o varios capitulos	-0.06	0.04	0.07	0.06	0.19*	0.22*
mensajes no entendidos	-0.01	-0.02	-0.10	-0.16*	-0.15*	0.07
prop_mayusculas	0.06	0.05	0.02	-0.01	-0.09	0.17*
lexico	0.03	0.02	-0.00	-0.01	0.05	-0.19*

Tabla 5.3: Características No LIWC. Características con al menos p<0.01 en una variable. Se remarcan pares de características con p<0.01

5.1. Análisis como problema de Regresión

A continuación aplicamos los distintos modelos de Aprendizaje Automático para intentar obtener un modelo de Regresión.

Como se puede observar en la figura 5.4, a pesar de probar los modelos explicados en la sección 4.2, los resultados de la predicción de los distintos aspectos de la personalidad son bastante malos. Al parecerse mucho a los resultados del clasificador base (que siempre devuelve la media entre los usuario), salvo para la edad en el que se obtienen resultados significativos. El clasificador para la edad obtiene un error absoluto medio de 10.39 mientras el clasificador base obtiene un error absoluto medio de 13.07.

variable	modelo	MAE	MAE base
EX	[‘Power Transformer’ + ‘SKB k=40 f=f_regression’ + ‘SVR’]	8.80	8.90
EE	[‘Power Transformer’+ ‘SKB k=80 f=mutual_info_regression’+ ‘SVR’]	9.07	9.15
CO	[‘No-scaler’+ ‘SKB k=5 f=f_regression’+ ‘SVR’]	8.36	8.63
AG	[‘No-scaler’+ ‘SKB k=40 f=f_regression’+ ‘SVR’]	7.88	7.97
OP	[‘No-scaler’+ ‘SKB k=20 f=f_regression’+ ‘BayesianRidge’]	7.80	8.18
Edad	[‘Power Transformer’+ ‘SKB k=40 f=f_regression’+ ‘BayesianRidge’]	10.39	13.07

Tabla 5.4: Resultados del Aprendizaje Automático con Regresión. Mejor modelo para cada características y error medio junto con error medio base. SKB se corresponde con la clase SelectKBest de Sklearn

5.2. Análisis como problema de Clasificación

Por último aplicamos los modelos de Aprendizaje Automático de Clasificación para intentar obtener una clasificación en 3 categorías.

Como se puede observar en la figura 5.5 (Acc significa exactitud, es decir, el número predicciones correctas divido entre el número de predicciones y Acc base es la exactitud del clasificador base) los resultados no son demasiado buenos al igual que en los modelos basados en regresión.

La edad vuelve a producir resultados mejores que el resto de variables, obteniendo una precisión de 0.55, mientras que el clasificador base(que predice la clase con más elementos) obtiene una precisión de 0.34.

Sin embargo, aunque los resultados obtenidos no nos permitirían hacer un perfilado indirecto de personalidad, si vemos que la información recogido reduce la incertidumbre sobre la personalidad de un usuario. Es de esperar que con más tiempo de uso de la aplicación, y más datos recogidos, la capacidad de predicción aumente.

variable	modelo	Acc	Acc base
EX	[‘Standard Scaler’+‘SKB k=80 f=mutual_info_classif’+‘DecisionTree’]	0.39	0.3
EE	[‘None’+‘SKB k=20 f=f_classif’+‘KNeighborsClassifier’]	0.4	0.34
CO	[‘Standard Scaler’+‘SKB k=10 f=f_classif’+‘SVC’]	0.43	0.33
AG	[‘None’+‘SKB k=80 f=f_classif’+‘RandomForestClassifier’]	0.41	0.34
OP	[‘None’+‘Variance Threshold 0.1’+‘RandomForestClassifier’]	0.45	0.33
Edad	[‘Power Transformer’, ‘Variance Threshold 0.1’, ‘SVC’]	0.55	0.34

Tabla 5.5: Resultados del Aprendizaje Automático con Clasificación.Mejor modelo para cada características y exactitud junto con exactitud base. SKB se corresponde con la clase SelectKBest de Sklearn

6

Conclusiones

En este trabajo se ha diseñado e implementado la infraestructura software de un bot conversacional, así como la infraestructura necesaria para analizar los datos y encontrar correlaciones entre el uso del bot y la personalidad.

La mayor dificultad ha sido realizar un diseño satisfactorio a partir de los requisitos del proyecto, debido a la gran cantidad de opciones a la hora de implementar un bot conversacional. Esta dificultad se ha solventado mediante el desarrollo de varios prototipos que han ido concretando el diseño.

Después del periodo de recolección de datos, hemos podido observar que el uso del bot ha resultado ser una experiencia satisfactoria para los usuarios, aunque no se han podido obtener resultados significativos a la hora de predecir la personalidad de los mismos. Es probable que esto sea debido a una multitud de factores:

1.– La mayoría de estudios sobre modelado indirecto de la personalidad sobre texto no han logrado grandes resultados.

2.– La conversación con este bot es bastante pobre sintácticamente al estar compuesta por dos test que no dan opción a responder con frases completas.
Si se desarrollase un bot con capacidad de participar en una conversación de una manera más natural, probablemente se obtendrían mejores datos. Aunque ese tipo de tecnología está actualmente en investigación.

3.– La muestra es relativamente pequeña. Solo se ha podido estudiar en torno a 400 usuarios, debido al uso todavía limitado de los bots y de Telegram por parte de la población general. En un futuro se podría repetir con una muestra mayor.

Por otro lado, este trabajo consigue el objetivo de diseñar e implementar un bot conversacional con la infraestructura software necesaria para realizar el modelado indirecto, es decir, consigue el objetivo principal del trabajo.

Además consigue realizar el análisis de resultados mediante el uso de Aprendizaje Automático de manera semiautomática. Por lo tanto, desde el punto de vista de la Ingeniería informática, el trabajo es correcto y es probable que se necesiten más investigaciones conjuntas entre informáticos y psicólogos

para obtener resultados más satisfactorios.

Bibliografía

[1] S. C. Cloninger, *Teorías de la personalidad (3a. ed.)*. Naucalpan de Juárez: Pearson Educación, 2002.

[2] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," *The Journal of artificial intelligence research*, vol. 30, pp. 457–500, 2007.

[3] J. Golbeck, C. Robles, M. Edmondson, and K. Turner, "Predicting personality from twitter." IEEE, 2011, pp. 149–156.

[4] A. Ortigosa, R. M. Carro, and J. I. Quiroga, "Predicting user personality by mining social interactions in facebook," *Journal of computer and system sciences*, vol. 80, no. 1, pp. 57–71, 2014.

[5] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria, "Deep learning-based document modeling for personality detection from text," *IEEE intelligent systems*, vol. 32, no. 2, pp. 74–79, 2017.

[6] A. Mota Garrido, "SimulaciÓn y estudio de la personalidad a travÉs del lenguaje mediado por ordenador," *Drafts of Economic Inteligence*, vol. 1, no. 4, pp. 35–44, 2018.

[7] *Test de personalidad: aprende a realizarlos*. Madrid: Editorial CEP, S.L, 2011.

[8] N. Fernández Rouco, R. Ruiz Cobo, J. A. Del Barrio del Campo, A. Ibáñez García, I. Salcines Talledo, E. Santurde del Arco, and J. M. Sánchez Rodríguez, "Características de la personalidad según el género en universitarios españoles," *Revista internacional de psicología*, vol. 13, no. 2, pp. 1–23, 2014.

[9] A. Vigil-Colet, F. Morales-Vives, E. Camps, J. Tous, and U. Lorenzo-Seva, "Development and validation of the overall personality assessment scale (operas)," *Psicothema*, vol. 25, no. 1, p. 100, 2013.

[10] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010. [Online]. Available: https://doi.org/10.1177/0261927X09351676

[11] C. Wei, Z. Yu, and S. Fong, "How to build a chatbot: Chatbot framework and its capabilities," ser. ICMLC 2018. ACM, 2018, pp. 369–373.

[12] "Integrations | dialogflow documentation | google cloud," https://cloud.google.com/dialogflow/docs/integrations, 2020, [Online; accedido 08/08/2020].

[13] "Configure a bot to run on one or more channels - bot service | microsoft docs," https://docs.microsoft.com/en-us/azure/bot-service/bot-service-manage-channels?view=azure-bot-service-4.0, 2019, [Online; accedido 08/08/2020].

[14] "Integrate your amazon lex bot with ant messaging service," https://aws.amazon.com/es/blogs/machine-learning/integrate-your-amazon-lex-bot-with-any-messaging-service/, 2017, [Online; accedido 08/08/2020].

[15] “Azure bot service introduction - bot service | microsoft docs,” https://docs.microsoft.com/en-US/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-4.0, 2019, [Online; accedido 08/08/2020].

[16] “Configure the bot - amazon lex,” https://docs.aws.amazon.com/en_en/lex/latest/dg/gs2-create-bot-configure-bot.html, 2020, [Online; accedido 08/08/2020].

[17] “Quickstart: Build an agent | dialogflow documentation | google cloud,” https://cloud.google.com/dialogflow/docs/quick/build-agent, 2020, [Online; accedido 08/08/2020].

[18] “Fulfillment | dialogflow documentation | google cloud,” https://cloud.google.com/dialogflow/docs/fulfillment-overview, 2020, [Online; accedido 08/08/2020].

[19] “Webhook service | dialogflow documentation | google cloud,” https://cloud.google.com/dialogflow/docs/fulfillment-webhook, 2020, [Online; accedido 08/08/2020].

[20] “Dialogflow basics | dialogflow documentation | google cloud,” https://cloud.google.com/dialogflow/docs/basics, 2020, [Online; accedido 09/08/2020].

[21] P. Rodriguez, A. Ortigosa, and R. M. Carro, “Extracting emotions from texts in e-learning environments,” in *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*. IEEE, 2012, pp. 887–892.

[22] “Liwc | linguistic inquiry and word count,” http://liwc.wpengine.com/, 2021, [Online; accedido 07/02/2021].

Definiciones

Alexa bot conversacional desarrollado por Amazon. Funciona como asistente personal.

AWS Lambda servicio de Amazon para la computación en la nube orientado a eventos.

Azure Bot Service Framework de Microsoft para el desarrollo de bots conversacionales.

Big Five Modelo psicológico de la personalidad que la describe como una combinación de cinco rasgos(Extraversión, Estabilidad Emocional, Responsabilidad, Cordialidad, Apertura a la Experiencia)..

bot programa que realiza tareas repetitivas. En este ensayo usado como sinónimo de Bot conversacional..

bot conversacional Programa diseñado para interactuar con el uso mediante lenguaje natural, normalmente a través de una aplicación de mensajería.

Cortana Bot conversacional desarrollado por Microsoft para su sistema operativo Windows 10.

Dialogflow framework de Google para el desarrollo de bots conversacionales..

ELIZA bot desarrollado por en 1972 por Kenneth Colby para imitar a un psicólogo.

Facebook Red social enfocada a compartir imágenes.

Facebook Messenger Aplicación de Mensajería desarrollada por la empresa Facebook.

Flask framework para el desarrollo de aplicaciones web en Python. Incluye un servidor web para desarrollo..

framework Conjunto de programas, librerías y en ocasiones lenguajes para el desarrollo de un tipo específico de software..

Google Now Asistente personal desarrollado por Google.

Gunicorn Servidor web para aplicaciones web python. Se integra mediante la interfaz WSGI con frameworks como flask.

HBO Plataforma de streaming.

LUIS Módulo de Azure Bot Service para el reconocimiento de la intención del usuario..

Netflix Plataforma de streaming.

Neuroticismo Personalidad con tendencia a la negatividad.

NodeJS Framework para el desarrollo de servidores de aplicaciones web..

plataforma de streaming Plataforma web que permite la reproducción de video sin descargarlo previamente.

PostgreSQL Gestor de bases de datos SQL.

Psicoticismo personalidad con tendencia a no estar conforme con las normas sociales..

Python Lenguaje de programación de propósito general.

QnA Sistema diseñado para responder preguntas.

Siri Asistente personal desarrollado por Apple.

Skype Aplicación de videoconferencia.

SQLAlchemy librería que permite la conexión de programas Python con gestores de bases de datos SQL.

Tay bot para Twitter desarrollado por Microsoft y desconectado a las pocas horas al producir mensajes ofensivos.

Telegram Aplicación de mensajería para Android e iOS.

Twitter Red social caracterizada por la publicación de mensajes cortos..

Webhook método para ampliar el funcionamiento de una aplicación web o en el caso de este proyecto del bot conversacional. Funciona a través de peticiones HTTP POST con mensajes JSON.

Whatsapp Aplicación de mensajería para Android e iOS.

Acrónimos

AIML Artificial Inteligence Markup Language.

A.L.I.C.E. Artificial Linguistic Internet Computer Entity.

API Aplication Programming Interface.

EPQ-R Eysenck Personality Questionnaire - Revised.

JSON JavaScript Object Notation.

LIWC Linguistic Inquiry and Word Count.

MIT Massachusetts Institute of Technology.

NLP Natural Language Processing.

OPERAS Overall PERsonality Assessment Scale.

WSGI Web Server Gateway Interface.

XML eXtensible Markup Language.

Apéndices

A

Nombre de cada categoría del LIWC

1	Pronom	24	Inhib	47	Ocupa
2	Yo	25	Tentat	48	Escuela
3	Nosotros	26	Certeza	49	Trabajo
4	unomismo	27	Sentidos	50	Logro
5	Tu	28	Ver	51	Placer
6	Otro	29	Oír	52	Casa
7	Negación	30	Sentir	53	Deportes
8	Asentir	31	Social	54	TV
9	Artículo	32	Comu	55	Música
10	Prepo	33	Refotro	56	Dinero
11	Número	34	Amigos	57	Metafo
12	Afectiva	35	Familia	58	Relig
13	Emopos	36	Humanos	59	Muerte
14	Sentpos	37	Tiempo	60	Físico
15	Optimi	38	Pasado	61	cuerpo
16	Emoneg	39	Presente	62	Sexual
17	Ansiedad	40	Futuro	63	Comer
18	Enojo	41	Espacio	64	Dormir
19	Tristeza	42	Arriba	65	Asearse
20	MecCog	43	Abajo	66	Maldecir
21	Causa	44	Incl	67	Nonfl
22	Insight	45	Excl	68	Fillers
23	Discrep	46	Moción		

Tabla A.1: Nombres LIWC